UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/703,449	10/31/2000	Stepan Sokolov	SUN1P814/P5417	1902

22434 7590 04/08/2004

BEYER WEAVER & THOMAS LLP
P.O. BOX 778
BERKELEY, CA 94704-0778

EXAMINER	
KENDALL, CHUCK O	

ART UNIT	PAPER NUMBER
2122	8

DATE MAILED: 04/08/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/703,449	SOKOLOV ET AL.	
	Examiner	Art Unit	
	Chuck O Kendall	2122	

-- The MAILING DATE f this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 January 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the application filed 1/13/04.
2. Claims 1 – ~~28~~ 30 have been examined.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1 – 9,11 – 21, 29 and 30 are rejected under 35 U.S.C. 102(e) as being anticipated Augusteijn et al. USPN 6,292,883 B1 (hereinafter Augusteijn).

Regarding claims 1 & 21 Augusteijn anticipates a method of creating data structures, an object oriented programming environment (1:44 – 46, JAVA TM) suitable for use by a virtual machine to execute computer instructions, the method comprising:

converting a stream of commands and data associated with the commands into a pair of streams for use in the virtual machine, the pair of streams including a code stream that includes the commands and a data stream that includes the data associated with the commands in the code stream (7:3 –10 ,23 – 30, also see 12: 1 – 5 and 8 – 15, see selection data, conversion data and virtual instructions).

Regarding claims 2 & 12, a method as recited in claim 1, wherein the code stream includes only commands and the data stream includes only the data associated with the commands in the code stream (7:1 – 6).

Regarding claims 3,13, &18 a method as recited in claim 1, wherein the stream that is to be converted is a JAVA TM compliant bytecode stream, the code stream is a JAVA TM bytecode code stream that includes JAVA TM commands, and the data

stream is a JAVA TM bytecode data stream that includes the data associated with the JAVA TM commands in the JAVA TM bytecode code stream (13:28 – 30, note JAVA TM bytecode is the same as a virtual machine instruction as stated in prior art, and refer to, 7:3 –10, 23 – 30).

Regarding claim 4, a method as recited in claim 1, wherein said converting of said stream comprises:

writing a representation of a first command associated with a first instruction into a code entry of the code stream (2:63 – 3: 10, see table and microcode and storing for writing, also see 12:9, for storing selection data);

determining whether the first command has data associated with it (12: 13– 15, for associated see conversion data indicated by selection data); and

writing a representation of the associated data or a reference to a representation of the data associated with the first command into a first data entry of the data stream when the command has associated data (12: 1- 15, see conversion table for data entry or data stream).

Regarding claim 5, a method as recited in claim 4, wherein the stream that is to be converted is a JAVA TM bytecode stream, the code stream is a JAVA TM bytecode code stream that includes JAVA TM commands, and the data stream is a JAVA TM bytecode data stream that includes the data associated with the JAVA TM commands in the JAVA TM bytecode code stream (10:30 – 35, for writing data see "stored").

Regarding claim 6, a method as recited in claim 4, wherein said method further comprises:

not providing a data entry in the data stream for the command when the command does not have data associated with it (10:23 – 27, see " may use autonomous conversion" or " shared logic").

Regarding claim 7, a method as recited in claim 6, wherein said method further comprises:

writing a representation of second command associated with another instruction into a second code entry of the code stream (10:25 – 30, see for each of the virtual machines and fig. 3 shows 2 VM's,332 and 336);

determining whether the second command has data associated with it; and writing a representation of the associated data or a reference to a representation of the data associated with the second command into a second data entry of the data stream when the command has associated data (10:20 – 35).

Regarding claim 8, a method as recited in claim 7, wherein the stream that is to be converted is a JAVA TM bytecode stream, the code stream is a JAVA TM bytecode code stream that includes JAVA TM commands, and the data stream is a JAVA TM bytecode data stream that includes the data associated with the JAVA TM commands in the JAVA TM bytecode code stream (7:23 – 30).

Regarding claims 9 & 14, a method as recited in claim 8, wherein the command and data entries can each include a number of bytecodes in the JAVA TM bytecode stream, wherein the number of bytecodes is an integer, and wherein each byte code can be one or more bytes (13:25 – 30).

Regarding claim 11, in an object oriented programming environment a computer readable medium including computer program code generating computer executable commands and data associated with the computer executable commands suitable for use by a virtual machine and comprising:

computer program for receiving a first stream of virtual machine instructions that include virtual machine commands and data associated with the commands (11: 62 – 67);

computer program code for generating, from the first stream, a code stream having one or more virtual machine commands (7:1 – 15); and

computer program code for generating, from the first stream, a data stream having data associated with the one or more virtual machine commands (9: 13 – 37, see selection data and virtual machine instruction).

Regarding claim 15, a data structure as recited in claim 14, wherein the JAVA TM commands can be a load constant command, an invoke method command, a jump command, an instantiation command, or a get/put field command (6:30 – 45, see pointing by pointer / jump command, class loader and retrieve field).

Regarding claim 16, Augusteijn anticipates a method of executing computer instructions on a virtual machine, the method comprising:

fetching a command associated with a virtual machine computer instruction from a code stream (7:20 – 25);

determining whether the command has an associated parameter (10:47 – 55);

fetching from a data stream the associated parameter of the command when said determining determines that command has an associated parameter (10:47 – 55) ; and executing the command with the associated parameters after the associated parameter of the commands have been fetched (10:47 – 55, for command see conversion means).

Regarding claim 17, a method as recited in claim 16, wherein the method further comprises:

updating a pointer to the command stream (11:20 – 25 , see change of instruction pointer); and

updating a pointer to the data stream (10:30 – 40).

Regarding claim 19, see claim 14 for reasoning.

Regarding claim 20, is the method claim corresponding to claim 15 and is rejected using the same rationale, as in claim 15.

Regarding claim 29, Augusteijn discloses a method of executing a virtual machine instruction on a virtual machine, the method comprising:

wherein the virtual machine instruction has a code portion and one or more data portions the converting the virtual machine instruction into a pair of streams, the pair of representation of the code portion of the virtual machine instruction, and wherein the data stream provides data that is needed to execute the code stream (7:3 –10 ,23 – 30, also see 12: 1 – 5 and 8 – 15, see selection data, conversion data and virtual instructions);

reading the code portion from the codes stream (11:56 – 59);

directly accessing the data from the code stream (7:20 – 25); and

executing the virtual machine instruction using the data directly accessed form the code portion (8: 13 – 20). However, Augusteijn doesn't explicitly disclose

reading a virtual machine instruction from a first stream (11:56 – 59);

Regarding claim 30, a method as recited in claim 29, wherein said converting the virtual machine instruction is performed at load time when a class file that includes the virtual machine instruction is loaded in the virtual machine (FIG.3, 342 and 346, also see associated text).

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Augusteijn et al. USPN 6,292,883 B1 (hereinafter Augusteijn) as applied in claim 9, in view of Wahbe et al. USPN 6,151,618 (hereiafter Wahbe).

Regarding claim 10, Augusteijn discloses all the claimed limitations as applied in claim 9 above. Augusteijn doesn't explicitly disclose wherein the first and second entries of the code stream are adjacent to each other. However, Wahbe does disclose this feature (17:33 – 37). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Augusteijn and Wahbe because, associating or combining adjacent entries of code, "This ensures that the operand specialization technique will not compete with the opcode combination technique by further specializing an instruction before the combiner has a chance to consider a less-specialized version"(17:50 – 55).

10. Claims 22-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Augusteijn et al. USPN 6,292,883 B1 (hereinafter Augusteijn) as applied in claim 21, in view of Toutonghi et al. USPN 5,2920,720 (hereinafter Toutonghi).

Regarding claim 22, Augusteijn discloses all the claimed limitations as applied in claim 21, above. Augusteijn doesn't explicitly reading and processing the associated data from a Constant Pool when the command has an associated data. However, Toutonghi does disclose this feature (6:37 – 40). Therefore, it would have been obvious to one of ordinary skill in the art at the invention was made to combine Augusteijn and Toutonghi because, associating static data or constant data in a JAVA TM environment using a constant pool, would make associating the variables or data more efficient.

Regarding claim 23, a method as recited in claim 22, wherein said processing operates to determine a constant value associated with a JAVA TM Load Constant command (Toutonghi, 5:50-55, see verifier which checks form of the bytecodes, also see fig.2, 114, 100).

Regarding claim 24, a method as recited in claim 22, wherein said processing operates to determine a reference to a method invocation cell that includes information relating to a JAVA TM invoke method command (Toutonghi, fig.2, 112).

Regarding claim 25, a method as recited in claim 22, wherein said processing operates to determine the code stream offset and data stream offset associated with a JAVA TM jump command (Toutonghi, 3:10 – 15).

Regarding claim 26, a method as recited in claim 22, wherein said processing operates to process a Constant Pool associated with a JAVA TM instantiation command (Toutonghi, 3:5 – 15).

Regarding claim 27, a method as recited in claim 22, wherein said processing operates to process a Constant Pool associated with a JAVA TM Get/Put field command (Toutonghi,6:35 – 46 , for retrieve data field).

Regarding claim 28, a method as recited in claim 22, wherein said processing operates to process data associated with a JAVA TM load constant command, a JAVA TM invoke method command, a JAVA TM jump command, a JAVA TM instantiation

command, or a JAVA TM get/put field command (Toutonghi, 3:1 – 15, see method, table pointer (jump command), instance, see 6:35 – 46, for retrieve data field).

Response to Arguments

11. Applicant's arguments filed 01/13/2004 with regards to claims 1 – 29 have been fully considered but they are not persuasive to over come the previous rejection.

Argument (1), In response dated 01/13/2004, Applicant argues in claims 1, 11 and 21, that Augusteijn doesn't teach " converting a virtual machine instruction in a first stream into a pair of streams for use in the virtual machine".

Response (1), Contrary to Applicant's argument Examiner believes that Augusteijn does show this functionality as set forth above in claims, see Augusteijn 12: 1 – 5, " To enable conversion of the further virtual machine instructions into native instructions also conversions data is received", and 12:8 – 15, " To be able to select the appropriate conversions means during execution, also data selection data is stored in the processing unit associating each further virtual machine instruction with the conversion data." As disclosed here, Examiner understands converting between instructions (virtual instructions), requires a retrieval of the associated selection data, which is matched up (paired) with the virtual instruction during conversion.

Argument (2), Applicant also argues in claim 4, that Augusteijn, doesn't disclose " " writing a representation of a first command associated with a first instruction into a code entry of the code stream, determining whether the first command has data associated with it, and writing a representation of the associated data or a reference to

a representation of the data associated with the first command into a first data entry of the data stream when the command has associated data.".

Response (2), Regarding Applicant's arguments as set forth above in claims Augusteijn does discuss these limitations for e.g.:

" ...writing a representation of a first command associated with a first instruction into a code entry of the code stream..." see 12:9, for storing selection data; and

for

"...determining whether the first command has data associated with it...", also see 12: 13 – 15, for associated see conversion data indicated by selection data; and

for

"...writing a representation of the associated data or a reference to a representation of the data associated with the first command into a first data entry of the data stream when the command has associated data." see, 12: 1 – 15, see conversion table for data entry or data stream.

Argument (3), In claim 16 of Applicant's response, Applicant also argues that Augusteijn doesn't teach " fetching a command associated with a virtual machine computer instruction from a code stream, and fetching from a data stream".

Response (3), As set forth in claims Examiner believes this limitation to be equivalent to Augusteijn limitations as seen in 12: 1 – 15. Here Augusteijn shows fetching instructions as well as using selection data to retrieve and associate conversion data with the virtual instructions.

Correspondence Information

12. Any inquires concerning this communication or earlier communications from the examiner should be directed to Chuck O.

Kendall who may be reached via telephone at (703) 308-6608. The examiner can normally be reached Monday through Friday between 8:00 A.M. and 5:00 P.M. est.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached at (703) 305-4552.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-3900.

For facsimile (fax) send to 703-7467239 official and 703-7467240 draft

Chuck O. Kendall

Software Engineer Patent Examiner

Chameli C. Das

CHAMELI C. DAS
PRIMARY EXAMINER